

SANDIA REPORT

SAND98-2753

Unlimited Release

Printed December 1998

Effective Protocols for Mobile Communications and Networking

Brian Van Leeuwen, Juan Espinoza, Jr., and Peter Sholander

Prepared by

Sandia National Laboratories

Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (703) 605-6000
Web site: <http://www.ntis.gov/ordering.htm>

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01



SAND98-2753
Unlimited Release
Printed December 1998

Effective Protocols for Mobile Communications and Networking

Brian Van Leeuwen
Secure Communication Systems Department

Juan Espinoza Jr.
Software Technologies and Development Department

Peter Sholander
Advanced Networking Integration

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0449

Abstract

This report examines methods of mobile communications with an emphasis on mobile computing and wireless communications. Many of the advances in communications involve the use of Internet Protocol (IP), Asynchronous Transfer Mode (ATM), and ad hoc network protocols. However, many of the advances in these protocols have been focused on wired communications. Recently much focus has been directed at advancing communication technology in the area of mobile wireless networks. This report discusses various protocols used in mobile communications and proposes a number of extensions to existing protocols. A detailed discussion is also included on desirable protocol characteristics and evaluation criteria. In addition, the report includes a discussion on several network simulation tools that may be used to evaluate network protocols.

Acknowledgment

The authors thank Luis Martinez and Larry Tolendino of the Advanced Networking Integration Division and Bruce Mah of the Security & Networking Research Division at Sandia National Laboratories for their support of the ATM and PNNI research.

Contents

1	Introduction.....	1
2	Routing Protocols Performance Issues and Evaluation Considerations.....	1
2.1	Applications	1
2.2	Characteristics of Mobile Ad hoc Networks.....	2
2.3	MANET Routing Protocol Performance Issues	3
3	Performance Modeling and Simulation Tools	6
3.1	Simulation Requirements.....	7
3.2	Simulation Tools	8
3.2.1	COMNET III (CACI)	8
3.2.2	MODSIM III (CACI).....	9
3.2.3	Network Simulator - ns (version 2)	11
3.2.4	OMNeT++ Discrete Event Simulation System.....	12
3.2.5	OPNET (MIL3, Inc.)	13
3.2.6	PARSEC.....	15
3.3	Summary of Simulation Tools	17
4	Mobile Routers in Mobile IP	17
5	Some Ad Hoc Network Protocols	19
5.1	Destination-Sequenced Distance Vector (DSDV)	20
5.2	Temporally-Ordered Routing Algorithm (TORA).....	20
5.3	Dynamic Source Routing (DSR).....	20
5.4	Ad Hoc On-Demand Distance Vector (AODV)	21
5.5	Associativity-Based Routing.....	22
5.6	Distance Vector Bellman Ford.....	23
5.7	Protocol Summary	23
6	ATM Considerations for Mobile Ad Hoc Networking	25
6.1	Effects of Mobile ATM Switches on PNNI Peer Group Operation.....	25
6.2	Effects of User Mobility on Usage Parameter Control in WATM Networks.....	26
6.3	PNNI Routing Support for Ad Hoc Mobile Networking: A Flat Architecture.....	27
6.4	PNNI Routing Support for Ad Hoc Mobile Networking: A Multilevel Case	28
7	Conclusion	28
8	References.....	30

Intentionally Left Blank

1 Introduction

With recent advances in communication technology, it has become possible to provide additional capabilities in the area of mobile communications. Mobile communications applications include mobile computing and wireless communications. Many of the advances in communications involve the use of Internet Protocol (IP), Asynchronous Transfer Mode (ATM), and ad hoc network protocols. However, many of the advances in these protocols have been focused on wired communications. Recently much focus has been directed at advancing communication technology in the area of mobile wireless networks. Many of these mobile wireless networks are based on the IP suite and on recent advances in wireless ATM.

This report describes the areas of investigation supported by the Wireless ATM LDRD project. Areas of investigation include the current state of the mobile networking technology related to mobile IP, ATM, and ad hoc network protocols. The report includes discussions on the performance evaluation criteria and issues related to mobile networks. Discussions on mobile IP and ad hoc network protocols are also included. The final section presents proposed extensions to ATM protocols to enhance mobile networking. Specific areas of the ATM protocol suite were investigated and a number of reports were written that include discussions on issues surrounding mobility in ATM. Simulation tools used for network protocol performance evaluation are also discussed.

2 Routing Protocols Performance Issues and Evaluation Considerations

A number of projects at Sandia National Laboratories require the use of mobile networks. In general, many technical applications would benefit from the use of mobile networks. This report will focus its discussions on mobile ad hoc networking. An ad hoc network is a set of wireless mobile nodes dynamically forming a communication network without the use of any existing network infrastructure or centralized management. Mobile ad hoc networks support robust and efficient operation in mobile wireless networks by incorporating routing functionality into mobile nodes. These networks will consist of dynamic, random, multihop topologies with the capability of changing rapidly. It is expected that these links are bandwidth constrained.

This section of the report will discuss the desired characteristics of Mobile Ad hoc Networks (MANETs). Additional discussions will address issues related to design and evaluation of network control protocols with an emphasis on routing performance evaluation considerations.

2.1 Applications

There is an increasing need for dynamic self-configuring or dynamic ad hoc network technology. Current attention has been focused on mobile IP technology. However, many applications require highly adaptive mobile networks that can effectively manage multihop, ad hoc network clusters that can operate autonomously and dynamically

configure their network topologies. In many cases, these networks will also have a need to interface with semi-fixed network topologies such as the Internet.

Some MANET applications include networks for future military systems. One such system is the anti-personnel landmine replacement. The system may require man-in-the-loop control and therefore communications are required. Another application is the developing technologies of battlefield communications where soldiers have communication networks that have highly dynamic topology segments. In addition, commercial applications include mesh-based mobile networks that can be operated as an inexpensive alternative to cell based mobile network infrastructures. MANET technology can also provide a rapidly deployable and flexible method for establishing communications for emergency operations such as fire and safety rescue operations. These are a few examples of MANET applications; many other applications can be envisioned as the technology is developed.

2.2 Characteristics of Mobile Ad hoc Networks

A MANET consists of a group of nodes. These nodes are based on mobile platforms, typically have a wireless communication device, and may have some router functionality. The nodes may be located over some geographical area or may be located in some vehicle such as an airplane, ship, truck, or any other host system. The MANET is able to dynamically reconfigure its communications topology autonomously. It is expected that the system will operate in isolation in that communication between network nodes is all that is necessary for the group of nodes to perform its functions. In some cases the MANET may have gateway nodes that provide a long-haul communication capability or may provide access to a semi-fixed network such as the Internet.

The nodes that make up the MANET have a means to communicate with each other. This communication is typically done via a wireless radio frequency (RF) link. The nodes have wireless transmitters and receivers. The nodes communicate with each other via a RF link and may not have contact with all the other nodes in the network. This is the reason that some or all of the nodes of the MANET must have the capability to forward and route messages to the specified destination node. The node transmitters and receivers use antennas that may be omnidirectional or directional, which will have an impact on the nodes' ability to communicate with other nodes. An additional consideration for a MANET node is to provide a degree of RF transmitter power control. Transmitter power control will allow the node to adjust its transmitter power to a level that is just enough to reach its intended recipient whether it is the final destination or a routing node. This is a method of conserving battery life. At any given point in time; the node's position in the network, its transmitter and receiver coverage patterns, and any channel interference, will determine its ability to communicate in the network. It is the function of the ad hoc network to provide either direct connectivity or multihop connectivity between the nodes. The ad hoc topology will, in many cases, be changing over time as the nodes physically move or adjust their transmission and receiving parameters.

A number of the characteristics of the MANET can be described as follows¹:

- **Dynamic topologies:** Nodes are free to move arbitrarily; thus, the network topology, which is typically multihop, may change randomly and rapidly at unpredictable times. The communications may consist of bidirectional links or multiple unidirectional links.
- **Bandwidth-constrained, variable capacity links:** Wireless links will continue to have significantly lower capacity than their hardwired counterparts. In addition, the realized throughput of a wireless communication link is further reduced when one considers the effects of multiple access, fading, noise, and interference conditions. In many cases the data throughput is often much less than a radio's maximum transmission rate.

One effect of the relatively low to moderate link capacities is that congestion is typically the norm rather than the exception. In many applications the aggregate demand will likely approach or exceed network capacity frequently. As the mobile network is often simply an extension of the fixed network infrastructure, mobile ad hoc users will demand similar services. These demands will continue to increase as multimedia computing and collaborative networking applications rise.

- **Energy-constrained operation:** Some or all of the nodes in a MANET may rely on batteries or other exhaustible means for their energy. For these nodes, the most important system design criteria for optimization will likely be energy conservation.
- **Limited physical security:** Mobile wireless networks are generally more prone to physical security threats than are fixed-cable nets. The increased possibility of eavesdropping, spoofing, and denial-of-service attacks should be carefully considered. Existing link security techniques are often applied within wireless networks to reduce security threats. As a benefit, the decentralized nature of network control in MANETs provides additional robustness against the single points of failure of more centralized approaches.
- **Need for scalability:** The wide range of applications of a MANET suggests that scalability will be an important factor. As an example, military applications may deploy topologies with hundreds of nodes in a single routing area.

2.3 MANET Routing Protocol Performance Issues

In the development of routing protocols, a number of criteria should be used to measure the performance of the protocol. Following is a list of desirable attributes of a MANET routing protocol. The following performance issues include those being considered by the MANET Working Group.

- **Distributed operation:** This characteristic is an essential property of a MANET protocol. There cannot be a dependence on a single node for network operation since it is expected that nodes will join and leave the network on a frequent basis.
- **Loop-freedom:** A desirable characteristic of a MANET is operational efficiency, which includes both power consumption and resource use such as RF bandwidth. Given that the wireless media used for communication is a shared media, it is expected that under most circumstances additional loading to the wireless media will cause performance degradation.
- **Demand-based operation:** The intention of this effort is to develop a protocol that will dynamically configure the network connectivity with the limited resources available. It will be a more efficient use of resources if the routing algorithm were to base the connectivity on traffic patterns. Instead of assuming a uniform traffic distribution within the network (and maintaining routing between all nodes at all times), the routing algorithm should adapt to the traffic pattern on a demand or need basis. If this is done intelligently, it will utilize network energy and bandwidth resources more efficiently.
- **Security:** Under most circumstances a MANET will use a wireless media for its communications link. The wireless media suffers from the possibility of eavesdropping, spoofing, denial-of-service attacks, and traffic analysis attacks. It may be relatively simple to snoop network traffic, replay transmissions, manipulate packet headers, and redirect routing messages, within a wireless network without appropriate security provisions. Security should be provided with some form of application layer, network layer, or link layer security. While these concerns exist within wired infrastructures and routing protocols as well, maintaining the "physical" security of the transmission media is harder in practice with MANETs. Sufficient security protection to prohibit disruption or modification of protocol operation is desired.
- **"Sleep" mode operation:** The MANET nodes are expected to operate autonomously with no tether to supply power. The nodes will have batteries to supply its power. As a result of energy conservation, or some other need to be inactive, nodes of a MANET may stop transmitting and/or receiving for arbitrary time periods. A routing protocol should be able to accommodate sleep periods without overly adverse consequences. This property may require close coupling with the link-layer protocol through a standardized interface.

The following is a list of quantitative metrics that can be used to assess the performance of any routing protocol.

- **End-to-end data throughput:** Data throughput is always important when assessing the performance of any network. Protocol performance evaluations will include statistical measures of data forwarding performance.

- **End-to-end data delay:** In addition to data throughput, end-to-end data delay should also be evaluated. Statistical measures of data routing performance are also important with the data delay.
- **Route Acquisition Time:** An additional time that contributes to the end-to-end data delay is the route acquisition time. The route acquisition time is the time required to establish routes when requested. This is of particular concern with “on demand” routing algorithms. An “on-demand” routing algorithm does not store routing information prior to the message routing request. The route is determined when a message must be transmitted.
- **Efficiency:** In a MANET the control messages and the data usually share the same wireless channel. In most cases the wireless channel capacity is limited and excessive control traffic will impact the data forwarding performance. Thus, the amount of control messages or overhead that is communicated over the channel will impact overall performance. Two ratios that provide information on the efficiency of the protocol are:
 - Average number of data bits transmitted/data bit delivered: This ratio measures the efficiency of delivering data within the network.
 - Average number of control bits transmitted/data bit delivered: This ratio measures the efficiency of the protocol in using control overhead to delivery data packets. This ratio should include not only the bits in the routing control packets, but also the bits in the header of the data packets. In other words, anything that is not data is control overhead, and should be counted in the control portion of the algorithm.

In addition to the above quantitative metrics that address many of the MANET specific issues consideration must also be given to general networking characteristics. Following are a number of network characteristics that should be measured.

- **Network size:** Measured in the number of nodes that make up the network.
- **Network connectivity:** The number of neighboring nodes that a node can communicate with.
- **Topological rate of change:** The speed with which a network's topology can change and the network protocol can maintain connectivity.
- **Link capacity:** Effective link data rate measured in bits/second, after accounting for losses due to multiple access, coding, framing, etc.
- **Fraction of unidirectional links:** How effectively does a protocol perform when one considers the presence of unidirectional links?

- **Traffic patterns:** With the emphasis on network efficiency and basing traffic patterns on a demand or need basis, how effective is a protocol in adapting to potentially non-uniform or bursty traffic patterns?
- **Mobility:** Many MANET applications will have real-time nodal movement occurring during network operation. Consideration must be given to the protocol's ability to perform during spatial topological changes. In these cases, how will the metric be applied to a spatially dynamic network and what is the most appropriate model for simulating node mobility in a MANET?
- **Fraction and frequency of sleeping nodes:** Many of the applications that are being considered for the MANET protocol must survive on battery power for extended periods of time and their functions must be duty cycled. Thus it is expected that nodes will be in a “sleep” mode for periods of time. How does a protocol perform in the presence of sleeping and awakening nodes?

The above discussion addresses the characteristics and performance criteria that should be considered when developing protocol for a MANET. The discussion includes the characteristics and performance criteria used in the MANET Working Group. It is believed that most required characteristics have been addressed; however, some specific applications may require additional characteristics or may have reduced dependence on any of the above mentioned criteria. The developed protocol should function effectively over a wide range of networking situations. Applications include small networks with minimal communication traffic to large networks with large amounts of communication traffic that may use the shared channel at near capacity. In addition, the MANET will have varying degrees of nodal mobility. The nodal mobility may be as minimal as a single topology configuration to a continuously changing topology.

An important criterion of MANETs is that they operate in a wireless communication environment, which suggests scarcity of resources. These resources include limited link bandwidth, limited power supply, and limited nodal capability.

This part of the report addressed issues that must be addressed as a MANET protocol is being developed. The specific applications must also be considered during the development. However, the intent is to develop a protocol that can be used for many of the MANET applications without detailed customization to that protocol. There are many engineering tradeoffs that must be considered, and these tradeoffs must be evaluated based on some consistent performance criteria.

3 Performance Modeling and Simulation Tools

An ad hoc network is a set of wireless mobile nodes dynamically forming a communication network without the use of any existing network infrastructure or centralized management. Due to the limited transmission range of wireless network interfaces and battery power constraints, multiple network hops may be needed for one node to exchange data with another across the network. In recent years, a variety of

routing protocols targeted specifically for this environment have been developed, but little performance information on each protocol and no realistic performance comparison between them has been available. In addition, any new protocol under development needs to undergo rigorous test and evaluation before being deployed. Computer simulation is the key to testing these proposed protocols to determine their performance characteristics and to evaluate them against their target environments and against other potentially competing protocols.

Computer simulation is the discipline of designing a model of an actual or theoretical physical system, executing the model on a digital computer, and analyzing the execution output. To understand reality and all of its complexity, we must build artificial objects and dynamically act out roles with them. Computer simulation is the electronic equivalent of this type of role-playing and it serves to drive synthetic environments and virtual worlds. Within the overall task of simulation, there are three primary sub-fields: model design, model execution, and model analysis. There are many software tools that support all three areas of computer simulation.

3.1 Simulation Requirements

A computer network uses two levels of protocols:

- A high-level routing protocol to route messages to their destinations.
- A low-level media access control (MAC) protocol to arbitrate and control access to the shared communication medium.

Both types of protocols are being studied in earnest in both the commercial and academic fields to advance the state of telecommunications for the military, private, and public sectors.

The simulation tool environment must be capable of simulating the following characteristics of a wireless ad-hoc network:

- Routing protocol
- MAC protocol
- Node mobility
- A realistic physical layer including a radio propagation model supporting propagation delay, capture effects, and carrier sense
- Radio network interfaces with properties such as transmission power, antenna gain, and receiver sensitivity

3.2 Simulation Tools

There are many commercial and academic simulation tools and toolkits available to the network administrator, network designer, and protocol designer today. However, to support this project's specific needs, we needed a tool that could be programmed to emulate a wireless ad-hoc network and support both routing and MAC protocols under investigation.

The following sections describe several commercial and academic tools and toolkits we investigated for use in our simulation efforts. Many of the descriptions are from web sites and promotional materials. None of the tools met all of our requirements in terms of cost, programmability, ease of use, training, extensibility, graphics support, and data analysis. However, the survey suggests that several of the tools used together would support most of our needs.

The following tools were investigated and descriptions follow:

- COMNET III (CACI, Inc.)
- MODSIM III (CACI, Inc.)
- Ns (UC-Berkley)
- OMNeT++ (University of Budapest)
- OPNET (Mil3, Inc.)
- PARSEC (UC-Los Angeles)

3.2.1 COMNET III (CACI)

COMNET III, a graphical, off-the-shelf package, lets you quickly and easily analyze and predict the performance of networks ranging from simple LANs to complex enterprise-wide systems. COMNET III builds on the success of CACI's thirty-five years of experience in simulation technology².

COMNET III supports a building-block approach where the blocks are objects you are familiar with in the real world. You start with a library of network objects with one COMNET III object representing one or more real world objects. The COMNET III object's parameters are easily adjusted to match the real-world object.

COMNET III's object-oriented framework gives you the flexibility to try an unlimited number of "what if" scenarios. An easy-to-understand animated picture of the network configuration you have selected will support your recommendations—no programming required.

As communication networks expand in scope and diversity, the planning challenge for network managers has become immense. Network monitoring tools that collect traffic data, discover and display network topologies, and display alarms are in wide use. There is a great need, however, for network design and planning tools that allow new networks or proposed changes to existing networks to be quickly evaluated before committing to a

course of action. This need for performance prediction is met through simulation analysis.

In simulation analysis, the computer traces out in detail the implications and consequences of a proposed network or change. As a result, simulation is more realistic and simulation results are easier to understand than with other forms of analysis.

COMNET III provides you with simulation-based network planning. Drawing on COMNET III's extensive and growing library of objects, you lay out your network using the icon drag-and-drop technique or, alternately, import the current topology from your Network Management System.

Simulation follows immediately—you see an animated picture and dynamic graphs while the simulation is running. You can interrupt the simulation to change parameters and take down links and nodes on the fly— with the effect of such changes immediately apparent.

COMNET III answers your questions about link and node utilization, message delays, and congestion points in large networks. It can answer questions about the performance of the current network topology based on historical or projected traffic loads.

COMNET III can answer questions about the performance of proposed networks or new applications:

- With historical network loads, you have a valuable network technology learning tool. By setting up a network model and experimenting with alternatives, you learn about networking technologies, especially the interactions between network components and applications.
- With projected loads, you see whether your current network can handle expected traffic growth or how much time you have before you need to upgrade the network.
- With proposed networks and applications, you can predict network performance.

3.2.2 MODSIM III (CACI)

MODSIM III is a complete development environment. The Simulation Layer, Graphics Editor, Compilation Manager, Object Manager, and the Debugging Manager provide the environment required for the successful development of advanced models. C++ interoperability provides access to C++ libraries³.

3.2.2.1 Simulation Layer

Complex systems are often composed of hundreds or even thousands of objects such as airplanes, ships, messages, parts, and so on. To ensure accurate modeling, an enormous number of state variables and behaviors for system components must be managed. Because all the necessary simulation management features are built-in, instead of being accessed through call statements, your model is compact and readable, making it much easier to develop, maintain, and enhance.

3.2.2.2 Graphics Editor

The MODSIM III graphics environment allows easy access to animation, dynamic presentation graphics, and user interface toolkits using the SIMDRAW graphics editor to simplify construction. Animation is produced by drawing objects using SIMDRAW and then animating them within a MODSIM III program. Operations such as scaling, rotating and positioning can be performed. The animated objects or images can have subcomponents that move along with the whole object, but can be manipulated individually.

Presentation graphics or graphs such as pie charts, level meters, bar graphs, etc. also are created using SIMDRAW and then used within a MODSIM III program by asking them to plot values. The visual appearance of the graph is updated automatically.

Access to user interface toolkits or forms allows the creation of a graphical user interface whose appearance conforms to the style of the system MODSIM III is running on. For example, on a Unix workstation Motif dialog boxes are used and on a Windows system dialog boxes and pull down menus are produced.

All elements of the MODSIM III graphics library are portable, which means if you take a MODSIM III graphics program which runs on one system and move it onto a new system it will run, without modifying any code. The forms will change their appearance to conform to the new system they are running on but the images and graphs will appear the same as they did on the previous system.

3.2.2.3 Compilation Manager

The MODSIM III compilation manager automatically determines which modules have been edited since the last compilation and recompiles only those modules and any other modules that depend on them. No "make" files are required.

3.2.2.4 Object-Oriented Development

The model of the system you are studying is described as a collection of interacting objects. Each object module describes the object behaviors, called methods. In addition, the characteristics of the object are contained in its variables. Everything the object knows is in its variables; and everything it can do is in its methods. This structure maps well to real world objects that comprise the system being modeled. Inheritance allows new objects to be defined as a special case of one or more general classes, reducing coding effort by automatically inheriting the method and variable definitions of the general class.

3.2.2.5 Object Manager

MODSIM III provides a browsing tool for objects. The Object Manager provides a concise representation of complex objects including aggregation of fields and methods and inheritance diagrams. You can see all the attributes and capabilities that an object has either defined or acquired through inheritance.

3.2.2.6 *Debugging Manager*

Because models are large complex computer programs, extensive support for detecting and isolating both simulation and programming errors is crucial to success.

Extensive run-time checking of object referencing, invalid parameters, array bounds, and memory use, for example, are invaluable aids to timely software development. Errors are detected when they are first incurred. A run-time error automatically drops you into debugging mode; allowing you to see where the error occurred and letting you examine variables. A traceback shows you the calling chain that led to the current method or procedure, so you can browse up and down the execution stack examining the sequence of procedure and method calls that preceded the error. The debugger supports a wide range of capabilities that are essential to interactive symbolic debugging and, because it has special knowledge of simulation constructs, it can display the pending list of events, simulation time, and memory usage information.

3.2.2.7 *C++ Interoperability*

C++ is gaining in popularity for general programming. By generating C++ and providing a simple way to interface with existing C++ modules, MODSIM III gives you convenient access to the large and growing base of existing C++ libraries.

3.2.3 *Network Simulator - ns (version 2)*

Ns is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols. Ns began as a variant of the REAL network simulator in 1989 and has evolved substantially over the past few years. The University of California at Berkeley and the VINT project developed ns. While it provides substantial support for simulating TCP and other protocols over conventional networks, it provides no support for accurately simulating the physical aspects of multi-hop wireless networks or the MAC protocols needed in such environments. Berkeley has recently released *ns* code that provides some support for modeling wireless LANs, but this code cannot be used for studying multi-hop ad-hoc networks. It does not support the notion of node position; there is no spatial diversity (all nodes are in the same collision domain), and it can only model directly connected nodes⁴.

Ns is an event-driven network simulator. An extensible simulation engine is implemented in C++ that uses MIT's Object Tool Command Language, OTcl (an object oriented version of Tcl) as the command and configuration interface. A previous version of the simulator, ns version 1, used the Tool Command Language, Tcl as the configuration language. The current version still supports simulation scripts written in Tcl meant for the ns version 1 simulator.

The simulator is invoked via the ns interpreter, an extension of the vanilla otclsh command shell. An OTcl script defines a simulation. The scripts use the Simulator Class as the principal interface to the simulation engine. Using the methods defined in this class, a network topology is defined, traffic sources and sinks are configured, the simulation is invoked, and the statistics are collected. By building upon a fully functional language, arbitrary actions can be programmed into the configuration.

The first step in the simulation is to acquire an instance of the Simulator class. Instances of objects in classes are created and destroyed in ns using the new and delete methods.

A network topology is realized using three primitive building blocks: nodes, links, and agents. The Simulator class has methods to create/ configure each of these building blocks. Nodes are created with the node simulator method that automatically assigns a unique address to each node. Links are created between nodes to form a network topology with the simplex-link and duplex-link methods that set up unidirectional and bi-directional links respectively. Agents are the objects that actively drive the simulation. Agents can be thought of as the processes and/or transport entities that run on nodes that may be end hosts or routers. Traffic sources and sinks, dynamic routing modules and the various protocol modules are all examples of agents. Agents are created by instantiating objects in the subclass of class Agent, Agent/type, where type specifies the nature of the agent.

Once the agents are created, they are attached to nodes with the attach-agent Simulator method. Each agent is automatically assigned a port number unique across all agents on a given node (analogous to a TCP or UDP port). Some types of agents may have sources attached to them while others may generate their own data. For example, you can attach ``ftp" and ``telnet" sources to ``TCP" agents but ``constant bit-rate" agents generate their own data. Sources are attached to agents using the attach-source and attach-traffic agent methods.

3.2.4 OMNeT++ Discrete Event Simulation System

OMNeT++ is an object-oriented modular discrete event simulator. The name itself stands for Objective Modular Network Testbed in C++. OMNeT++ has its roots in OMNeT, a simulator written in Object Pascal by Dr. György Pongor⁵.

The simulator can be used for modeling communication protocols, computer networks and traffic modeling, multi-processors and distributed systems, and several other systems where the discrete event approach is suitable. OMNeT++ supports animation and interactive execution.

3.2.4.1 Modeling Principles

An OMNeT++ model consists of modules that communicate via message passing. The active modules are termed simple modules; their code executes quasi in parallel (simple modules are implemented with lightweight processes or co-routines.) Simple module code is written in C++ using the simulation class library. Messages can contain arbitrarily complex data structures. Messages can be sent either directly to their destination or through a gate (in this case, the message travels along a predefined path, though a series of connections).

Simple modules can be grouped into compound modules and so on; the number of hierarchy levels is not limited. There is a separate topology description language for defining compound module types. NED (NEtwork Description) has a simple syntax, yet

it is very powerful: for example, it is possible to define a hypercube network with the size as parameter.

3.2.4.2 More Than Just a Library

OMNeT++ contains much more than just the simulation library. It has several components that can help in many phases of a simulation project: model design, debugging/tracing, interactive execution, visualization, prescribing multiple runs, selecting seeds, visualization of the results etc.

OMNeT++ supports parallel execution of large simulations. Currently, Conservative PDES and Statistical Synchronization can be used as a synchronization method. To make use of parallel execution, the model is to be partitioned to several segments that will run on different hosts or processors. The topology description language supports model partitioning in a flexible way.

Message sending between different partitions of the model is transparent: the user does not have to be aware that the destination of the message is on a different processor. A message can contain arbitrarily complex data structures; they are transferred transparently, even between different architectures. Debugging of remotely executing parts is supported by log files and by relaying textual output messages to the console. OMNeT++ uses the PVM3 (Parallel Virtual Machine Version 3) library for communication between hosts. PVM is portable and it is widely used in university and research environment.

3.2.4.3 Supported Platforms

With OMNeT++, you are not tied to one platform or another: the most convenient platform is a Unix system with some GNU software, but you can also use it on Win95/NT or even DOS (with DJGPP or BC3.1). The program is available in source -- it can be studied, experimented with and freely modified; or it can be the starting point of your own developments.

OMNeT++ was/is currently developed at the Technical University of Budapest, Department of Telecommunications (BME-HIT). Information and source code on OMNeT++ is available at the following URL:

3.2.5 OPNET (MIL3, Inc.)

OPNET is a comprehensive software environment for modeling, simulating, and analyzing the performance of communications networks, computer systems and applications, and distributed systems. MIL 3's clients are the leading communications organizations worldwide in both commercial and government sectors⁶.

OPNET is used as a decision support tool to provide insight into the performance and behavior of existing or proposed networks, systems, and processes. Specific tools are included with each OPNET license that assist users through the following phases of the modeling and simulation cycle:

3.2.5.1 Model Building and Configuration

- Network Editor - define or change network topology models
- Node Editor - define or change data flow (systems architecture) models
- Process Editor - define or change control flow (behavioral logic) models

3.2.5.2 Running Simulations

- Simulation Tool - define and run simulation studies using models constructed with the OPNET Editors
- Interactive Debugging Tool - interact with running simulations

3.2.5.3 Analyzing Results

- Analysis Tool - display and compare statistical results
- Animation Viewer - watch dynamic behavior of models during simulation runs

3.2.5.4 Modeling Methodology and Model Hierarchy

To provide useful data, network models must combine accurate descriptions of topology, data flow, and control flow. Since no single paradigm of visual representation is ideally suited for all three of these model types, OPNET utilizes separate model formats for each. To lend structure and discipline to the overall model, OPNET models fit together in a hierarchical fashion.

OPNET Network Models define the position and interconnection of communicating entities, or nodes. Each node is described by a block structured data flow diagram, or OPNET Node Model, which typically depicts the interrelation of processes, protocols, and subsystems. Each programmable block in a Node Model has its functionality defined by an OPNET Process Model which combines the graphical power of a state-transition diagram with the flexibility of a standard programming language and a broad library of pre-defined modeling functions.

The OPNET Network Editor graphically captures the topology of a communication network. Networks consist of node and link objects, which are assembled and parameterized via pop-up dialog boxes. To create node objects, users select node types from a library of example and user-defined models. Each OPNET Node Model has a specific set of attributes that used to configure it.

Network models can be constructed in a dimensioned workspace with a user-selected grid. Since users may define node abstractions, OPNET Network Models may represent LAN's, MAN's, WAN's, on-board vehicular networks, or any combination thereof. Sub-network objects can be used to structure an unlimited topology hierarchy within any Network Model. Users can load a customized cartographic background to define a physical context for wide-area networks (cartographic features are obtained from the CIA World Data Bank II).

The OPNET Node Editor graphically captures node architectures, which are diagrams of data flow between modules typically representing hardware and software subsystems. Module types include processors, queues, traffic generators, receivers, and transmitters.

Processors are general modules that provide complete flexibility in protocol and algorithm specification. The functionality of processor and queue objects is defined using OPNET Process Models.

Instances of OPNET Node Models are used to populate OPNET Network Models. The OPNET Process Editor uses a powerful state-transition diagram approach to support specification of any type of protocol, resource, application, algorithm, or queuing policy. States and transitions graphically define the progression of a process in response to simulated events. Within each state, general logic can be specified using a library of over 350 pre-defined functions. The full flexibility of the C programming language is also accessible. As with other OPNET editors, users can construct entirely new process models, or modify those provided by MIL 3.

Once a set of OPNET Network, Node, and Process Models are fully defined, users can run simulation studies based on them via the OPNET Simulation Tool. Users can see dynamic model behavior (at all three levels) in the OPNET Animation Viewer, and plot statistical performance measurements based on simulation studies in the OPNET Analysis Tool.

The OPNET Analysis Tool provides a graphical environment that allows users to view and manipulate data collected during simulation runs. Standard and user-specified probes can be inserted at any point in a model to collect statistics. Simulation output collected by probes can be displayed graphically, viewed numerically, or exported to other software packages. First and second order statistics on each trace as well as confidence intervals can be automatically calculated. OPNET supports the display of data traces as time-series plots, histograms, probability density and cumulative distribution functions, and scattergrams. Graphs (as with models at any level in the OPNET modeling hierarchy) may be output to a printer or saved as bitmap files to be included in reports or proposals.

3.2.6 *PARSEC*

PARSEC (for PARallel Simulation Environment for Complex systems) is a C-based discrete-event simulation language. It adopts the process interaction approach to discrete-event simulation. An object (also referred to as a physical process) or set of objects in the physical system is represented by a logical process. Interactions among physical processes (events) are modeled by timestamped message exchanges among the corresponding logical processes⁷.

One of the important distinguishing features of PARSEC is its ability to execute a discrete-event simulation model using several different asynchronous parallel simulation protocols on a variety of parallel architectures. PARSEC is designed to cleanly separate the description of a simulation model from the underlying simulation protocol, sequential or parallel, used to execute it. Thus, with few modifications, a PARSEC program may be executed using the traditional sequential (Global Event List) simulation protocol or one of many parallel optimistic or conservative protocols.

In addition, PARSEC provides powerful message receiving constructs that result in shorter and more natural simulation programs. Useful debugging facilities are available. A front-end for visual specification of simulation models and a runtime output display and visualization environment are currently being designed.

3.2.6.1 PARSEC Primitives

PARSEC adopts the process interaction approach to discrete-event simulation. An object (also referred to as a PP for physical process) or set of objects in the physical system is represented by a logical process or LP. Interactions among PPs (events) are modeled by message exchanges among the corresponding LPs. We first describe the PARSEC primitives to define processes and the interprocess communication primitives and subsequently indicate how they are used to describe events.

- **Entities** — A PARSEC program is a collection of entity definitions and C functions. An entity definition (or an entity type) describes a class of objects. An entity instance, called simply an entity, represents a specific object in the physical system. Every PARSEC program must have a driver entity. This entity initiates execution of the simulation program and serves essentially the same purpose as the main function in C.
- **Messages** — Entities communicate with each other using buffered message passing. Every entity has a unique message buffer; asynchronous send and receive primitives are provided to respectively deposit and remove messages from the buffer.
- **Events** — Each event in a discrete-event simulation model simulates some activity of interest in the physical system and may involve one or more objects. Every event is associated with a time stamp that indicates the time at which the corresponding event occurs in the system. Execution of a discrete-event simulation model requires that all events in the system be executed in their strict time stamp order. Since an event is modeled by a message in a PARSEC program, each message carries a timestamp. By default, the timestamp placed on a message is the current (simulation) time of the entity that sends the message. A message may be explicitly timestamped with a future time as described later.

3.2.6.2 Parallel Simulation

A PARSEC model can be executed using sequential or parallel simulation algorithms. The sequential algorithm typically uses an ordered data structure called the global event list that stores all events that are generated in the system in their timestamp order.

For parallel execution, each entity in the model is mapped to a specific processor. Each processor, say P, has its own event list which stores the events for the entities that are mapped to P. Parallel synchronization algorithms have been defined to ensure that events in the event lists stored on different processors are executed in their global timestamp order. Enforcing this requirement, referred to as the causality constraint, is the central problem in efficient execution of parallel simulations. Two primary approaches have been suggested to solve the synchronization problem: conservative and optimistic.

3.2.6.3 Platforms

PARSEC has been implemented on a variety of sequential workstations and laptop machines, on networks of workstations and on scalable MPP platforms like the distributed memory IBM SP2 and on symmetric multiprocessor (SMP) platforms like the shared memory SPARCstation 1000. It has been used for the parallel simulation of a number of applications in diverse areas including queuing networks, VLSI designs, parallel programs, mobile wireless multimedia networks, ATM networks, and high-speed communication switches and protocols.

3.3 Summary of Simulation Tools

In the author's opinion, the following tools would warrant further investigation for use in the development of performance simulations of wireless ad-hoc networks:

Tool	Pros	Cons
COMNET III	Graphics support, multi-platform support	Very expensive (>\$40K), no parallel support, network performance only, not programmable
MODSIM III	Graphics support, multi-platform support	Very expensive (>\$40K), no parallel support
ns	Graphics support, multi-platform support, free for noncommercial use	UNIX port to DOS, not mainstream
OMNeT++	Graphics support, Parallel support, free for noncommercial use	UNIX port to DOS, not mainstream
OPNET	Graphics support, multi-platform support	Very expensive (>\$40K), no parallel support, difficult to use
PARSEC	Parallel support, multi-platform support, free for noncommercial use	No graphics support

4 Mobile Routers in Mobile IP

Applications for mobile connectivity to the Internet have seen advances in "mobile IP" technology. Mobile IP technology is directed at providing a roaming host connectivity to the Internet. This technology provides a means for a roaming host, also called a nomadic host, to maintain connectivity to the Internet as its geographical location changes. The Internet is based on a known fixed address domain space. Fixed nodes have an address associated with the node and messages are routed to these nodes based on the address. However, a nomadic host is not connected to the Internet at a static location. The nomadic host may be physically connected to the Internet or may be connected via a

wireless link. The nomadic nature of the connection is the capability of the node to move to other locations on the Internet, even to different subnets⁸.

Current means of supporting nomadic hosts involves address management. A method based on IP utilizes a Home Agent (**HA**) and a Foreign Agent (**FA**). The **HA** has a fixed location on the Internet and thus a fixed Internet address. The **HA** is aware of the location of the associated nomadic host and can route messages to the nomadic host via the foreign agent. The foreign agent provides a connection to the Internet for the nomadic host.

Mobile IP depends on preexisting routing protocols operating within the fixed network. The approach is inefficient because messages are first sent to the **HA** which then must immediately forward to the **FA**. Thus other options for providing nodal mobility must be considered.

Mobile IP⁹ does have provisions for mobile routers. To review, Mobile IP has two methods, specified in RFC 2002, for allowing nodes to roam beyond their home networks. The two methods use Home Agents (**HAs**) and Foreign Agents (**FAs**) that function as surrogates for nodes that have left their home networks. Each node that will implement Mobile IP has a long-term IP address associated with its home network. This long-term IP address does not change under either method of implementing Mobile IP.

In the first method, a roaming node registers with the **FA** of the visited IP subnet and obtains a care-of-address for that subnet. The roaming node then registers that subnet care-of-address with the **HA** of its Home network. Once the roaming node has successfully registered with the **FA** and the **HA**, the mobile node can then send and receive datagrams. Outgoing IP datagrams can be directly routed to the destination IP address, while incoming datagrams are intercepted by the **HA** on the mobile node's Home network. The **HA** encapsulates the datagram in an IP datagram addressed to the care-of-address. The **FA** receives the "tunneled" datagram and delivers it to the mobile node's long-term IP address. Hence, incoming and outgoing IP datagrams may take different network paths.

Consideration was given to applying these techniques to mobility of ATM. However, our research identified reasons that this may not work. These different network paths will complicate the ATM QoS and ABR flow control calculations. There are two other difficulties with this method. First, it can cause congestion at the node's **HA**. Second, incoming datagrams may take sub-optimal routes. For example, a New York-based roaming node that is visiting California could receive datagrams, even those that originated in California, via its New York-based **HA**.

The second method assigns a co-located care-of-address to the mobile node. This temporary care-of-address is a proper address for the visited subnet, which is acquired through some external means such as DHCP. As in the previous method, the care-of-address is registered with the mobile node's **HA**. Sending and receiving datagrams still use a tunnel, but this time the tunnel terminates at the mobile node. This method has two

advantages. First, it removes the **FA** from the communications path. Second, it provides the outgoing datagrams with an originating IP address that is native to the visited subnet. These two features are important where **FAs** have not been deployed or where IP routers flag non-standard addresses as evidence of IP address spoofing or other anti-social behavior — and hence drop those IP packets.

Mobile IP¹⁰ permits mobile routers and mobile networks. A mobile router may function as the **FA** for mobile hosts visiting its subnetwork. The mobile router might communicate through its own **HA-FA** surrogates. Therefore, communications to a mobile node visiting a mobile router's subnetwork could involve two **HA-FA** pairs and recursive IP tunneling. This situation is bound to result in suboptimal datapaths. Also, the ATM cell-sequence and route-splitting problems make this a dubious model for mobile ATM switches.

5 Some Ad Hoc Network Protocols

A number of protocols that are currently in use and protocols that have been developed and are under consideration will contribute to the development of a MANET. In addition, the Internet Engineering Task Force (IETF) MANET Working Group is considering the various developments in protocol development. The MANET working Group has a near term goal of standardizing a MANET protocol.

A number of areas of investigation and technology development supported by the working group is as follows:

- Protocols that support effective operation over a wide range of mobile networking applications.
- Protocols and algorithms that support node discovery that allow newly arriving nodes to join existing MANETs.
- Protocols and algorithms that support route maintenance that allow nodal movement and topology changes in MANETs.
- Protocols and algorithms that support efficient response to topology changes and changing traffic demands thus maintaining efficient routing paths.
- Protocols and algorithms that support communication security in a shared media environment.

A number of ad hoc network protocols with various design choices have been proposed. Following is a list of these protocols:

- DSDV: Destination-Sequence Distance Vector
- TORA: Temporally-Ordered Routing Algorithm
- DSR: Dynamic Source Routing
- AODV: Ad Hoc On-Demand Distance Vector
- ABR: Associativity-Based Routing
- BFR: Bellman-Ford Routing

The following sections will give a brief overview of each of the above mentioned routing protocols.

5.1 Destination-Sequenced Distance Vector (DSDV)

DSDV¹¹ is a hop-by-hop distance vector routing protocol requiring each node to periodically broadcast routing updates. The key advantage of DSDV over traditional distance vector protocols is that it guarantees loop-freedom. Each DSDV node maintains a routing table listing the “next hop” for each reachable destination. DSDV tags each route with a sequence number and considers a route **R** more favorable than if **R** has a greater sequence number, or if the two routes have equal sequence numbers but **R** has a lower metric. Each node in the network advertises a monotonically increasing even sequence number for itself.

When a node **B** decides that its route to a destination **D** has broken, it advertises the route to **D** with an infinite metric and a sequence number one greater than its sequence number for the route that has broken (making an odd sequence number). This causes any node **A** routing packets through **B** to incorporate the infinite-metric route into its routing table until node **A** hears a route to **D** with a higher sequence number.

5.2 Temporally-Ordered Routing Algorithm (TORA)

TORA¹² is a distributed routing protocol based on a “link reversal” algorithm¹³. It is designed to discover routes on demand, provide multiple routes to a destination, establish routes quickly, and minimize communication overhead by localizing algorithmic reaction to topological changes when possible. Route optimality (shortest-path routing) is considered of secondary importance, and longer routes are often used to avoid the overhead of discovering newer routes.

The actions taken by TORA can be described in terms of water flowing downhill towards a destination node through a network of tubes that models the routing state of the real network. The tubes represent links between nodes in the network, the junctions of tubes represent the nodes, and the water in the tubes represents the packets flowing towards the destination. Each node has a height with respect to the destination that is computed by the routing protocol. If a tube between nodes **A** and **B** becomes blocked such that water can no longer flow through it, the height of **A** is set to a height greater than that of any of its remaining neighbors, such that water will now flow back out of **A** (and towards the other nodes that had been routing packets to the destination via **A**).

5.3 Dynamic Source Routing (DSR)

DSR¹⁴ uses source routing rather than hop-by-hop routing, with each packet to be routed carrying in its header the complete, ordered list of nodes through which the packet must pass. The key advantage of source routing is that intermediate nodes do not need to maintain up-to-date routing information in order to route the packets they forward, since the packets themselves already contain all the routing decisions. This fact, coupled with

the on-demand nature of the protocol, eliminates the need for the periodic route advertisement and neighbor detection packets present in other protocols.

The DSR protocol consists of two mechanisms: Route Discovery and Route Maintenance. Route Discovery is the mechanism by which a node **S** wishing to send a packet to a destination **D** obtains a source route to **D**. To perform a Route Discovery, the source node **S** broadcasts a ROUTE REQUEST packet that is flooded through the network in a controlled manner and is answered by a ROUTE REPLY packet from either the destination node or another node that knows a route to the destination. To reduce the cost of Route Discovery, each node maintains a cache of source routes has learned or overheard, which it aggressively uses to limit the frequency and propagation of ROUTE REQUESTs.

Route Maintenance is the mechanism by which a packet's sender **S** detects if the network topology has changed such that it can no longer use its route to the destination **D** because two nodes listed in the route have moved out of range of each other. When Route Maintenance indicates a source route is broken, **S** is notified with a ROUTE ERROR packet. The sender **S** can then attempt to use any other route to **D** already in its cache or can invoke Route Discovery again to find a new route.

5.4 Ad Hoc On-Demand Distance Vector (AODV)

AODV¹⁵ is essentially a combination of both DSR and DSDV. It borrows the basic on-demand mechanism of Route Discovery and Route Maintenance from DSR, plus the use of hop-by-hop routing, sequence numbers, and periodic beacons from DSDV.

When a node **S** needs a route to some destination **D**, it broadcasts a ROUTE REQUEST message to its neighbors, including the last known sequence number for that destination. The ROUTE REQUEST is flooded in a controlled manner through the network until it reaches a node that has a route to the destination. Each node that forwards the ROUTE REQUEST creates a *reverse route* for itself back to node **S**. When the ROUTE REQUEST reaches a node with a route to **D**, that node generates a ROUTE REPLY that contains the number of hops necessary to reach **D** and the sequence number for **D** most recently seen by the node generating the REPLY. Each node that participates in forwarding this REPLY back toward the originator of the ROUTE REQUEST (node **S**), creates a *forward route* to **D**. The state created in each node along the path from **S** to **D** is hop-by-hop state; that is, each node remembers only the next hop and not the entire route, as would be done in source routing.

In order to maintain routes, AODV normally requires that each node periodically transmit a HELLO message, with a default rate of once per second. Failure to receive three consecutive HELLO messages from a neighbor is taken as an indication that the link to the neighbor in question is down. Alternatively, the AODV specification briefly suggests that a node may use physical layer or link layer methods to detect link breakage to nodes that it considers neighbors.

When a link goes down, any upstream node that has recently forwarded packets to a destination using that link is notified via an UNSOLICITED ROUTE REPLY containing an infinite metric for that destination. Upon receipt of such a ROUTE REPLY, a node must acquire a new route to the destination using Route Discovery as described above.

5.5 Associativity-Based Routing

Developed at Cambridge University, Associativity-Based Routing (ABR)¹⁶ is a protocol that is designed for a conference-size ad-hoc mobile network environment. Routes are established on demand. The uniqueness of this scheme is the route selection criteria. Route is selected based on nodes' associativity. By “associativity” or “affinity” we refer to the spatial, temporal, and connection relationship of a mobile host with its neighbors. We measure associativity by recording the number of control beacons received by a node from its neighbors. For example, let each mobile host have a transmission/reception range of 10m in diameter and let there be two mobile hosts **A** and **B**.

Assuming both **A** and **B** are not in radio connectivity with each other initially but each sends a control beacon to signify its presence once every second. If **A** is migrating at 2 m/s and it starts to enter **B**'s radio cell and move through it diagonally, then both **A** and **B** record at most five beacons each. Hence, this is the associativity threshold. If five beacons are recorded, then one can regard that the other mobile host is migrating past it and this is viewed as being associatively unstable.

Otherwise, if the mobile host is moving but is constantly within the radio coverage of its neighbors, then more than five beacons will be recorded and hence the node is regarded as being associatively stable. Note that associativity has an interlocking characteristic since a node's association stability with its neighbors depends on the mobility profile of the neighbors. By picking nodes in a route that has associativity counts/ticks that indicate association stability, the route will have a long-lived characteristic. Thus, in ABR, a route is composed of nodes that are “associatively stable” stayed within the transmission range of immediate upstream and downstream node for a long period of time. This could result in a route with non-shortest path, but the route can be maintained with less chance of having to perform route recovery. The detailed algorithm for route selection in ABR can be found in [16].

ABR does not require alternate route information for each source-destination pair. Instead, only the destination node of a route selects the best route. This prevents nodes from having stale routing information. ABR is free from loops, deadlocks, and packet duplicates. When a route needs to be reestablished, rather than the source invoking a new route establishing process, the immediate upstream of a node that violated the associativity performs a localized query. This makes the route recovery process much faster compared to other protocols. The following shall elaborate further on: (a) route discovery, (b) route reconstruction, and (c) route removal.

5.6 Distance Vector Bellman Ford

The distance vector Bellman Ford Routing (BFR) protocol was developed originally to support routing in the ARPANET. An existing implementation of it used to support routing in today's Internet is known as RIP (Routing Internet Protocol)¹⁷. It is a table-driven routing protocol, i.e.; each router constantly maintains an up-to-date routing table with information on how to reach all possible destinations in the network. For each entry, the next router to reach the destination and a metric to the destination are recorded. The metric can be the hop distance, total delay, or cost of sending the message. Each node in the network begins by informing its neighbors about its distance to all other nodes. The receiving nodes extract this information and modify its routing table if any route information is revised (the best route is switched to a different route or the metric to the destination is changed). The node uses the following formula to calculate the best route:

$$D(i; j) = \min_k [d(i; k) + D(k; j)]$$

where $D(i; j)$ is the metric of shortest path from node i to node j , $d(i; k)$ is the cost of traversing directly from node i to node k , and k is one of the neighbors of node i . After recomputing the metrics, nodes pass their own distance information to their neighboring nodes again. After a while, all nodes/routers in the network have a consistent routing table to all other nodes. This protocol is not appropriate for large networks and suffers from the “count-to-infinity” problem.

RIP uses a time-triggered (periodic, about 30sec interval) and event-triggered (every link changes or router failures) routing updates to inform all nodes in the network so that the routing tables can be updated. An update message contains the destination ID and its metric to the destination. Mobility can be viewed as link changes and/or router failures and in a mobile network environment, many event-triggered routing updates occur, in addition to the time-triggered updates. This therefore results in excessive inefficient usage of the limited wireless bandwidth.

5.7 Protocol Summary

The above protocols each have their pros and cons, each protocol is based on different assumptions and intuitions of the network. Recently a couple of research reports, [18][19] have made some performance comparisons of these protocols.

The first research report¹⁸ compared the DSDV, TORA, DSR, and AODV protocols. In summary the research concluded that these protocols cover a range of design choices, including periodic advertisements vs. on-demand route discovery, use of feedback from the MAC layer to indicate a failure to forward a packet to the next hop, and hop-by-hop routing vs. source routing. This research simulated each protocol in ad hoc networks of 50 mobile nodes moving about and communicating with each other, and presented the results for a range of node mobility rates and movement speeds.

Each of the protocols studied performed well in some cases yet had certain drawbacks in others. DSDV performs quite predictably, delivering virtually all data packets when node mobility rate and movement speed are low, and failing to converge as node mobility increases. TORA, although the worst performer in our experiments in terms of routing packet overhead, still delivered over 90% of the packets in scenarios with 10 or 20 sources. At 30 sources, the network was unable to handle all of the traffic generated by the routing protocol and a significant fraction of data packets were dropped. The performance of DSR was very good at all mobility rates and movement speeds, although its use of source routing increases the number of routing overhead bytes required by the protocol. Finally, AODV performs almost as well as DSR at all mobility rates and movement speeds and accomplishes its goal of eliminating source routing overhead, but it still requires the transmission of many routing overhead packets and at high rates of node mobility is actually more expensive than DSR.

A second research report¹⁹ compared the performance of the ABR and the DSR protocols. ABR is a protocol for conference-size ad hoc networks with mobile hosts. Routes are initiated by the source and are set up based on demand. The destination selects a route based on the association stability of nodes in the route and also considers fair distribution of route relaying functions among mobile hosts. The route reconstruction process exploits the advantage of locality of neighboring mobile hosts to quickly construct alternate and even shorter routes.

The researchers evaluated the performance of ABR via simulation. They showed that the reconstructed route can be shorter than its original route and the number of localized query operations is less than its worst possible value. In addition, as the route-neighboring factor is increased, the probability of a localized query success is also increased. They compared the performance of ABR with Bellman-Ford Routing and Dynamic Source Routing. Simulation results revealed that the BFR incurs extensive bandwidth and computation overhead, yielding inferior performance when compared to ABR and DSR on-demand routing protocols. They also observed that ABR has a better throughput, smaller delay, and fewer control overheads than DSR. Employing different route selection metrics and localizing the route recovery processes make ABR appropriate for the multihop mobile wireless environment.

However, the above mentioned research did not consider two important performance metrics when comparing the protocols. The two overlooked metrics are related to latency and are as follows:

- Route acquisition time
- End-to-end data delay

The issue of route acquisition latency may be a potential negative consequence of on-demand routing algorithms. This metric would be very useful in comparing various on-demand routing algorithms, as well as comparing on-demand and proactive approaches.

Data delivery latency is one of the most important performance metrics as it applies to all routing algorithms. Along with data throughput, it is fundamental to comparing routing algorithm performance. Without this metric, a realistic assessment of routing algorithm performance cannot be done.

6 ATM Considerations for Mobile Ad Hoc Networking

A number of studies were done during this project related to mobility with Asynchronous Transfer Mode (ATM). There are a number of interesting problems in mobility with ATM. Mobility management has two basic components; location management and handoff. Location management routes incoming calls to a mobile end-system's, or mobile switch's current location. Handoff then maintains ongoing ATM virtual circuit (VC) connectivity as the mobile's position subsequently changes.

Mobile switches in ATM are receiving much attention. Applications include both mobile switches with attached fixed networks (i.e. commercial airliners and military command centers) and also ad hoc network topologies. The ATM Forum's Private Network-Network Interface (PNNI) has an emphasis on switch mobility. However, the PNNI protocol does not support the dynamic nature of ad hoc networks.

Traditional routing protocols use distance vector or link-state vector routing algorithms. These algorithms assume a slowly varying network topology such that a coherent network view can be propagated throughout the network by periodic neighbor-to-neighbor broadcasts. Network topology changes occur due to congestion and node failure. Node adjacency is usually pre-defined by the network designer. Ad hoc networks break these assumptions since they are much more dynamic. Node adjacency is based upon reachability in a time-variant broadcast medium such as wireless links. The PNNI framework has problems with these issues.

This LDRD project investigated a number of ATM and PNNI protocol issues. A number of PNNI extensions were developed and proposed to the ATM Forum. The Wireless Asynchronous Transfer Mode (WATM) part of the LDRD has three parts. The first is the effect of mixed wired and wireless networks on standard Internet applications, such as web browsing, file transfer and video conferencing. The second focus is mobility management. This project has examined protocol extensions, to standard link-state routing protocols (such as Private Network-Network Interface (PNNI)), that support moveable ATM switches. DoD applications include mobile command centers. DOE applications include temporary test and monitoring networks. The last focus area is the interaction between user mobility and the current ATM traffic management procedures. Results indicate that traffic contract enforcement may be problematic for Variable Bit Rate (VBR) services such as MPEG-2 video²⁰.

6.1 Effects of Mobile ATM Switches on PNNI Peer Group Operation

The WATM part of the project has produced four major deliverables. The first was SAND Report 97-0911C on the effects of mobile ATM (Asynchronous Transfer Mode)

switches on PNNI (Private Network-Network Interface) peer group operation²¹. That report was presented, as ATM Forum contribution 97-0315, at the April 1997 meeting of the ATM Forum Technical Committee in Chicago. Current ATM Forum work is focused on endsystem mobility. However, some companies (SNL, IBM and Boeing) are also considering moveable ATM switches. Commercial applications include Local Area Networks (LANs) on airliners. DoD applications include mobile command centers. DOE applications include temporary test and monitoring networks.

The research presented in SAND Report 97-0911C discusses why, and how, mobile networks and mobile switches might be discussed during Phase 1 of the WATM standards process. Next, it reviews mobile routers within Mobile IP. That IP mobility architecture may not apply to the proposed mobile ATM switches. Finally, it discusses problems with PNNI peer group formation and operation when mobile ATM switches are present.

The report presents a brief overview of some the issues with WATM and how WATM is different from stationary ATM. Another section discusses Mobile IP and the two methods used for Mobile IP. Methods used in Mobile IP are compared with possible approaches for WATM.

PNNI is discussed. An overview of how it works; topics include Peer Group (PG) establishment and maintenance, peer group leader election, neighbor discovery, and topology database exchange. In addition, transient PG membership is discussed.

The report presents a promising approach with protocol extensions to the existing ATM routing protocol, PNNI. The report lays out the technical problems posed by those extensions. PNNI is a hierarchical link-state routing protocol. Peer Groups (PGs) are a key concept in PNNI. Individual switches obtain detailed information about their own PG's topology and summarized information about other PG's topologies. As such, a PG hierarchy is conceptually similar to the (country code)-(area code)-(central office code) routing paradigm in public phone networks. There are then three core technical problems. First, how does a moveable ATM switches learn about the other existing PGs? Second, how does that switch decide which PG to join ? Lastly, how does that switch seamlessly change PGs ?

6.2 Effects of User Mobility on Usage Parameter Control in WATM Networks

The second deliverable was SAND Report 97-1927C on the effects of user mobility on Usage Parameter Control (UPC) in Wireless Asynchronous Transfer Mode (WATM) Networks²². That report was submitted to the 1998 IEEE International Performance Computing and Communications Conference.

This report has identified a new network impairment for proposed Wireless ATM systems — namely the effects of user mobility on Usage Parameter Control (UPC) — that is not present in existing ATM networks and cellular networks. WATM networks pose new traffic management problems. One example is the effect of user mobility on

Usage Parameter Control (UPC). If the UPC algorithm resets after each handoff between wireless-cells, then users can cheat on their traffic contract. In brief, UPC is analogous to a digital filtering operation, with an associated convergence time. If a user changes wireless-cells before the UPC algorithm converges then UPC can not determine whether that user complied with their traffic contract. This report derives explicit relationships between a user's traffic parameters (Peak Cell Rate, Sustained Cell Rate and Maximum Burst Size), their transit time per wireless-cell, their maximum sustained cheating-rate and the Generic Cell Rate Algorithm's (GCRA's) Limit (L) parameter. It also shows that the GCRA can still effectively police Constant Bit Rate (CBR) traffic, but not some types of realistic Variable Bit Rate (VBR) traffic. Finally, it discusses three solutions for this new network impairment.

ATM call setup entails a "traffic contract". The user requests a certain quality of service (end-to-end delay, cell-loss and end-to-end jitter) from the network. The network may accept that request, subject to agreed-upon traffic parameters such as Peak Cell Rate (PCR) and Sustained Cell Rate. In that case, the network provides that QoS to the ATM cells that conform to those traffic parameters. The Generic Cell Rate Algorithm (GCRA) is the standard conformance test. It is a filtering algorithm that ensures, for example, PCR compliance to within some agreed-upon tolerance. As such, like any filtering algorithm, the GCRA has a finite settling time. Hence, in mobile networks, users might change wireless-cells before the GCRA settles. So, if the GCRA state resets after each wireless handoff then the network can't always determine conformance. Hence, malicious, or clueless, users can exceed their traffic contract with impunity. This can affect the QoS for other conformant users.

SAND Report 97-1927C gives explicit formulas for the relationships between a user's traffic parameters, their transit time per wireless-cell, their maximum sustained cheating rate and the GCRA's tolerance (or "limit parameter"). Results indicate that the GCRA can still police Constant Bit Rate (CBR) traffic, but not some types of realistic Variable Bit Rate (VBR) traffic. Finally, there may be no standards-based way to transfer UPC state during handoff. As such, some future public WATM networks may exclude those types of VBR traffic.

6.3 PNNI Routing Support for Ad Hoc Mobile Networking: A Flat Architecture

The third major deliverable was a report titled *PNNI Routing Support for Ad Hoc Mobile Networking: A Flat Architecture*. The report is ATM Forum contribution 97-1073²³. This contribution extends the Outside Nodal Hierarchy List (ONHL) procedures described in ATM Forum Contribution 97-0766. These extensions allow multiple mobile networks to form either an ad hoc network or an extension of a fixed PNNI infrastructure. This contribution covers the simplest case where the top-most Logical Group Nodes (LGNs), in those mobile networks, all reside at the same level in a PNNI hierarchy. Future contributions will cover the general case where those top-most LGNs reside at different hierarchy levels. This contribution considers a "flat" ad hoc network architecture – in the sense that each mobile network always participates in the PNNI hierarchy at the pre-configured level of its top-most LGN.

This contribution's protocol extensions allow multiple mobile networks to form either an ad hoc network or an extension of a fixed PNNI infrastructure. It covered the simplest case where the top-most LGNs, in those mobile networks, all reside at the same level in a PNNI hierarchy. Future contributions will cover the general case where those top-most LGNs reside at different PNNI hierarchy levels.

The proposed Mobile PGID Selection (MPGIDS) process allows a "flat" ad hoc network architecture – in the sense that each mobile network always participates in the PNNI hierarchy at the pre-configured level of its top-most LGN. The MPGIDS is a distributed, multiparty process that propagates its decision information via flooding. Hence, it may not meet all of the routing requirements of more general ad hoc network topologies [1]. However, it may be applicable to small ad hoc networks with low mobility rates. It may also apply to slow-moving gateway nodes that translate between PNNI and some ATM-centric version of MANET.

6.4 PNNI Routing Support for Ad Hoc Mobile Networking: A Multilevel Case

The forth major deliverable was a report titled *PNNI Routing Support for Ad Hoc Mobile Networking: A Multilevel Case*. The report is ATM Forum contribution 98-0072²⁴. This contribution extends the Outside Nodal Hierarchy List (ONHL) procedures described in ATM Forum Contributions 97-0766²⁵ and 97-0933. These extensions allow multiple mobile networks to form either an ad hoc network or an extension of a fixed PNNI infrastructure. A previous contribution (97-1073) covered the simplest case where the top-most Logical Group Nodes (LGNs), in those mobile networks, all resided at the same level in a PNNI hierarchy. This contribution covers the more general case wherein those top-most LGNs may reside at different PNNI hierarchy levels. Both of the SNL contributions consider "flat" ad hoc network architectures – in the sense that each mobile network always participates in the PNNI hierarchy at the pre-configured level of its top-most LGN.

The report showed that a merged mobile-network can recursively obtain fixed-network connectivity. However, it again points out a fundamental design tradeoff. There may be less network disruption if only the top-most LGN in the merged network joins the fixed-network. In that case, only one PGID changes. However, that simplification violates the design principle that each mobile network should make independent decisions. (That principle can not be ironclad though. Sections 2.3.2 and 2.3.4²⁶ of the Sholander report illustrated realistic topologies where a lower-level mobile-network's choice for its fixed network PG subsequently constrained the choices available to the other higher-level mobile-networks.)

7 Conclusion

This report presented various approaches and protocols used for routing information through dynamic network topologies or networks that have mobile nodes. The report focused on issues that mostly reside in the network and data link layers of the OSI

network architecture model. Network organization, link scheduling, and routing were addressed in the surveyed protocols. The lowest layer of the OSI network model, the physical layer, which addresses the media access control (MAC) is important for mobile applications and will be addressed in future work.

The report addressed the important characteristics of protocols for mobile radio networks. These characteristics are important when considering the various applications that these protocols may be applied. In addition, evaluation considerations were presented to determine the effectiveness of specific protocols.

Simulation tools are used to predict performance of network protocols and to provide a means to optimize protocol variables for specific applications. Various tools were discussed and their pros and cons were presented. Creating protocol models will be less complicated for the simulation tools that have a more extensive library of models. For the purposes presented in this report the tools with extensive libraries will be most applicable.

A number of protocols used in mobile networks were presented. The protocols used different approaches to solve similar problems and each has its pros and cons. This report presented a number of the most applicable protocols and presented their approach to solving the dynamic network topology difficulty.

This report also presented the work done to extend ATM to support mobile connectivity. A brief overview was presented on mobile IP and how IP can be made to support mobility. An overview of some of the issues with wireless-ATM was also presented. PNNI was discussed and proposed extensions were developed that would support mobility with ATM. Issues surrounding mobility of ATM switches and potential network impairments for proposed wireless-ATM systems were presented. A number of specific mobility impairment issues were identified and possible solutions have been proposed for these impairments.

Future work related to the research presented in this report includes evaluating and extending the performance of the developed routing protocols to operate with various media access techniques. The developed ad hoc protocols must be tailored to the application media access technique. In addition, the routing protocol should also consider the wireless media environment. The wireless media environment will determine the various RF channel distortions that include path loss, multi-path effects, noise, and dropped messages due to message collisions. Thus, future work will include understanding the RF channel and extending the proposed ad hoc protocols to operate in the non-idea environment.

8 References

-
- ¹ Corson, S., “Mobile Ad Hoc Networking: Routing Protocol Performance Issues and Evaluation Considerations,” Internet Draft, March 1998.
- ² World Wide Web site; <http://www.caciasl.com/comnetthree.html>, August 1998.
- ³ World Wide Web site; <http://www.caciasl.com/modsim.html>, August 1998.
- ⁴ World Wide Web site; <http://www-mash.CS.Berkeley.EDU/ns/>, August 1998, and Fall, K., and Varadhan, K., editors, *ns* Notes and Documentation.
- ⁵ World Wide Web site; <http://www.hit.bme.hu/phd/vargaa/omnetpp.htm>, August 1998.
- ⁶ World Wide Web site; <http://www.mil3.com/>, August 1998.
- ⁷ World Wide Web site; <http://PARSEC.cs.ucla.edu/>, August 1998.
- ⁸ Sholander, P., Tolendino, L., Martinez, L., “The Effects of Mobile ATM Switches on PNNI Peer Group Operation,” ATM Forum Contribution 97-0315, April 1998.
- ⁹ Perkins, C., “IP Mobility Support,” RFC 2002, October 1996.
- ¹⁰ Perkins, C., [9].
- ¹¹ Perkins, C., Bhagwat, P., “Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers,” *Proceedings of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, August 1994. A revised version of the paper is available from <http://www.cs.umd.edu/projects/mcml/papers/Sigcomm94.ps>.
- ¹² Park, V., Corson, S., “A highly adaptive distributed routing algorithm for mobile wireless networks,” In *Proceedings of INFOCOM'97*, pages 1405–1413, April 1997.
- ¹³ Park, V., Corson, S., “Temporally-Ordered Routing Algorithm (TORA) version 1: Functional specification,” Internet-Draft, draft-ietf-manet-tora-spec-00.txt, November 1997.
- ¹⁴ Broch, J., Johnson, D., Maltz, D., “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks,” Internet-Draft, draft-ietf-manet-dsr-00.txt, March 1998.
- ¹⁵ Perkins, C., “Ad Hoc On Demand Distance Vector (AODV) routing,” Internet-Draft, draft-ietf-manet-aodv-00.txt, November 1997.

-
- ¹⁶ Toh, C-K, "Associativity-Based Routing For Ad Hoc Mobile Networks," Wireless Personal Communications Journal, Special Issue on Mobile Networking and Computing Systems, Kluwer Academic Publishers, vol. 4, no. 2, pp. 103-139, March 1997.
- ¹⁷ Malkin, G., "RIP Version 2 - Carrying Additional Information," Internet Draft, draft-ietf-ripv2-protocol-v2-05.txt, June 1998.
- ¹⁸ Brock, J., Maltz, D., "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," To appear in Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98), October 25-30, 1998, Dallas, Texas.
- ¹⁹ Gerla, M., Lee S., Toh, C-K, "A Simulation Study of an Ad Hoc Routing Protocol for Multihop Mobile Wireless Networks," Computer Science Department, University of California, Los Angeles, August 1998.
- ²⁰ Sholander, P., "Network Surety Modeling for Wireless ATM Networks," Annual Report Summary, August 1997.
- ²¹ Sholander, P., Tolendino, L., Martinez, L., "The Effects of Mobile ATM Switches on PNNI Peer Group Operation," ATM Forum Contribution 97-0315, April 1998.
- ²² Sholander, P., Tolendino, L., Martinez, L., "The Effects of User Mobility on Usage Parameter Control (UPC) in Wireless ATM Systems," That report was submitted to the 1998 IEEE International Performance Computing and Communications Conference, September 1997.
- ²³ Sholander, P., Tolendino, L., Martinez, L., "PNNI Routing Support for Ad Hoc Mobile Networking: A Flat Architecture," ATM Forum Contribution 97-1073, November 1997.
- ²⁴ Sholander, P., Tolendino, L., Martinez, L., "PNNI Routing Support for Ad Hoc Mobile Networking: A Multilevel Case," ATM Forum Contribution 98-0072, February 1998.
- ²⁵ Dykeman, D., Iliadis, I., Scotton, P., "PNNI Routing Support for Mobile Networks," ATM Forum Contribution 97-0766, September 1997.
- ²⁶ Sholander, P. [24].